UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

A VEHICLE ROUTING IMPROVEMENT ALGORITHM - COMPARISON OF A 'GREEDY' AND A MATCHING IMPLEMENTATION FOR INVENTORY ROUTING

by

Moshe Dror*, Michael O. Ball**
and Larry Levy**

* Industrial Engineering and Management,
Ben-Gurion University of the Negev.
Visiting Professor, Centre de recherche sur les transports,
Université de Montréal, Summer 1984.

** College of Business and Management,
University of Maryland at College Park.

Université de Montréal
Centre de recherche sur les transports - Publication #364
Septembre 1984

Dépôt légal - troisième trimestre 1984

ABSTRACT

Given an initial solution to a vehicle routing problem - (VRP) - we present three heuristic route improvement schemes based on the concept of node interchange between different routes. Three algorithms are presented together with their computational performance when applied to an inventory routing problem for 12 consecutive weekly periods. All three procedures improve the initial solution constructed by a modified Clark and Wright algorithm by over 50% as measured by the change in the value of the objective function. In two of the algorithms a minimum weight matching problem is solved repetively.

1. INTRODUCTION

In this paper we present two routing improvement procedures and compare their performance. These two procedures were designed and implemented on a solution to an inventory routing problem - (IRP). The terminology 'inventory routing problem' is applied to a situation where a set of N customers (each with a different demand on each day) are resupplied from a central depot and the objective is to minimize the annual distribution costs while attempting to ensure that no customer runs out of the commodity at any time (the IRP is described in detail in [1], [2], [7], [8], [9], [10], [11]). In [7] and [8] a solution procedure for the IRP is described in which the problem is first reduced to a much shorter time horizon problem, i.e., to a problem for a weekly period called the "planning period".

In the process of problem reduction from an annual time base to a weekly timebase (for details see [7], [9]) two subsets of customers were selected and two sets of cost coefficients were constructed. The first subset of customers denoted by M consists of those customers which must be replenished during the given planning period. The second subset of customers denoted by M' is also selected for a given planning period. The customers in M' do not have to be replenished during the planning period. We assume that no customer will require more than one replenishment during any given planning period. For each customer $i \in M$ a finite sequence of cost coefficient C_{ik}, $k=1,2,\ldots, T_i$ (T_i could be an integer from 1 to 5) is constructed such that $C_{ik} > 0$ and $C_{ik} > C_{i,k+1}$ for all i and k. The cost coefficients C_{ik} represent a cost associated with assigning customer $i \in M$ for replenishment on day k. For each customer $i \in M'$ a finite sequence of cost coefficients g_{ik}, $k=1,5,\ldots,5$ is constructed such that $g_{ik} < 0$ and $g_{ik} > g_{i,k+1}$ for all i and k. The cost coefficients g_{ik} represent the cost

($-g_{ik}$ represents the savings) associatd with assigning customer $i \in M'$ for replenishment on day k.

Our solution to the inventory routing problem over an annual time base consists of a sequence of consecutive weekly solutions to the IRP as defined over a weekly planning period. When solving an IRP over a given planning period, we construct a feasible solution to the problem, i.e. a set of daily routes such that each in its total demand not exceeding the capacity of the vehicle, and such that the set of routes assigned on a given day to a vehicle do not exceed the time constraint on the length of the work day. All the customers which belong to the subset M are included in the routes constructed to be replenished on some day during the planning period; in addition a subset of customers from M' are included in the same routes.

After a feasible solution to the problem is obtained we attempt to improve the quality of this solution via a route improvement scheme. The route improvement scheme is designed to examine a given feasible solution and to search for favorable trade-offs when interchanging customers' positions on a route and between different routes. It represents a seperate optimization stage when solving the IRP. In attempts to exploit the costs related to the two characteristics of a customer in a solution :

(a) The temporal parameter (i.e., customer's demand level depends on what day he is scheduled for replenishement)

(b) The spatial parameter (i.e., the customer's location relative to other customers in the solution)

In addition, we form one artificial route that includes all the customers from the subset M' that are not assigned for replenishment during the planning period. Denote that subset of customers (and the artificial route) by

NM'. A customer that belongs to the subset NM' can be inserted into any other route but only customers from M' - NM' can be interchanged with a customer from NM'.

The goal of this improvement stage is to reduce the total distribution cost for the planning period as reflected by our objective function, while maintaining route capacity constraints. The two competing algorithms that attempt to accomplish that are based on what is called the Node Interchange Procedure. Before describing in detail the Node Interchange Prodecure, we clarify some conceptual differences between this procedure and arc interchange heuristics. (The most commonly known examples of the latter are the two-opt and r-opt procedures. A detailed description of two-opt and r-opt algorithms can be found in Lin [12] and Lin and Kernighan [13], and Christofides and Eilon [6].)

2. Comparison Between Node Interchange and Arc Interchange Methods

An early analysis of arc interchange heuristics to improve a given Traveling Salesman tour was presented in a paper by Lin [12]. In this paper Lin introduced the concept of r-optimality. A given Traveling Salesman tour is r-optimal if it is impossible to obtain a tour with smaller cost by replacing a set of r of its arcs by any other set or r-arcs. An r-optimal tour is also r'-optimal for $r' < r$.

An r exchange is a transformation of one tour to another tour by exchanging r-arcs in the first tour by a different set of r-arcs. Figure 2.1 illustrates a two-exchange transformation. v_{ij} denotes the direct travel cost between nodes i and j.

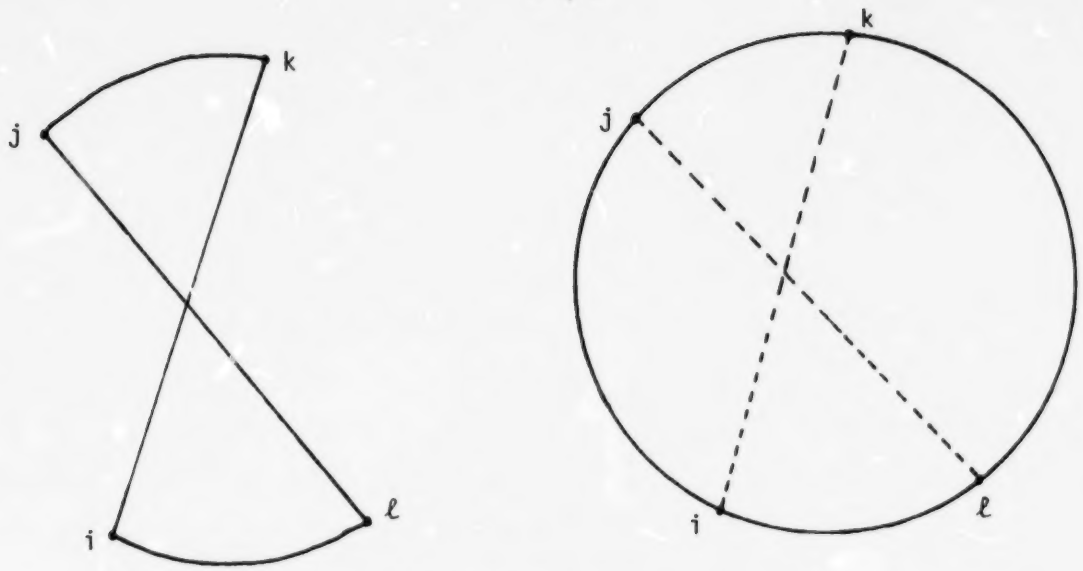

Figure 2.1 : Two-exchange transformation.

The r-exchange transformation changes only the sequence which orders a given set of nodes. It is designed for a single Traveling Salesman tour. In the case of the two-opt procedure a single savings term is given by

$$OS(i,j;\ k,\ell) = v_{ij} + v_{k\ell} - v_{ik} - v_{j\ell}$$

The Node Interchange Procedure is a two-node rather than two-arc interchange heuristic. In addition, unlike the two-opt heuristic, it is not limited to a single traveling salesman tour. (In order to apply the two-opt heuristic to a given m-traveling salesman tour a transformation of the m-tour to an equivalent 1-tour is necessary.) Given $m > 1$ tours, we say that the m tours are two swap-optimal if it is impossible to obtain an improved m-tour solution by exchanging the positions of one or two nodes. We illustrate a single one-node interchange operation in Figure 2.2 and a single two-node interchange operation in Figure 2.3. We define in a similar fashion an r swap-optimal set of routes.

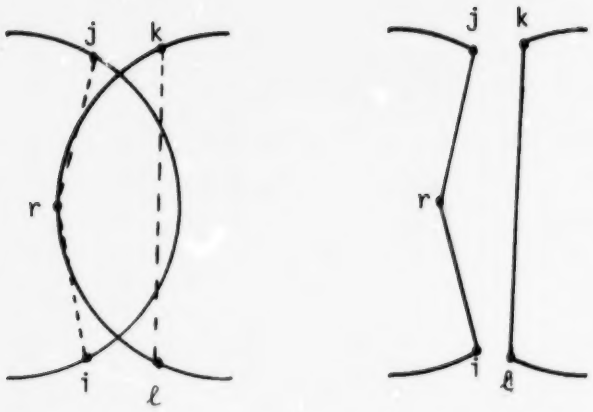

Figure 2.2 : One-node interchange.

The savings term for the one-node interchange transformation is the following :

$$OS(i,j;\ k,r,\ell) = v_{ij} + v_{kr} + v_{r\ell} - v_{ir} - v_{rj} - v_{k\ell}$$

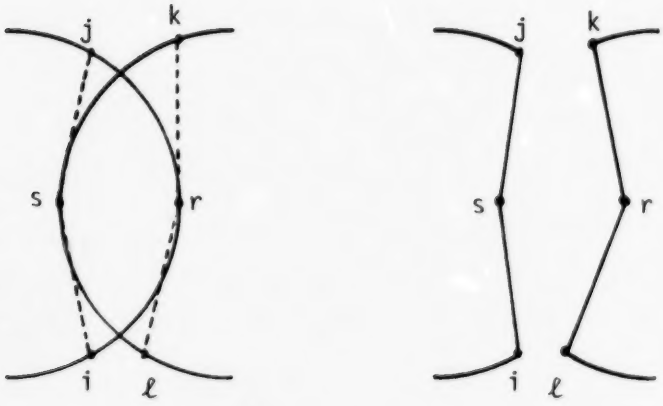

Figure 2.3 : Two-node interchange.

The savings term for the two-node interchange transformation is the following :

$$OS(i,s,j;\ k,r,\ell) = v_{jr} + v_{ri} + v_{ks} + v_{s\ell} - v_{js} - v_{si} - v_{kr} - v_{r\ell}$$

If the two-node interchange is applied to a single route, then it is equivalent to a four-arc exchange. If one-node interchange is applied to a single route then it is equivalent to a three-arc exchange. Lin in [12] found that for the TSP the four-arc exchange heuristic did not find noticeably better solutions than the three-opt solution. The strength of the node-interchange procedure is its applicability to an m-tour solution for $m > 1$.

For a given solution to a VRP, the candidate nodes for an interchange must have a strictly positive savings term and must satisfy the individual tour constraints as well.

3. The Calculation of the OS-(saving). Terms for the VRP Solution

Assume that the number of vehicles available each day is W. The number of routes per vehicle per day is U. Denote by P_k^{hr} the subset of customers that constitute the route r for the vehicle h on day k. The subset of customers assigned on the k^{th} day equals the union of P_k^{hr} over h and r, i.e., $\bigcup_{h,r} P_k^{hr}$. We denote that subset by P_k; k = 1,...,ND. Given a feasible set of routes for the whole planning period, in order to transform this set of routes into a set of routes that are two-swap-optimal, a number of different node interchange operations may be required. We list and

and describe all the different node-interchanges below.

(1) One node interchange between routes on the same day (customer switching routes).

(2) One node interchange between routes on different days (customer switching routes and days of delivery).

(3) One node interchange between a customer on a regular route and the artificial route (deletion). The artificial route is the route that passes through all the customers in M' not scheduled for delivery during the planning period.

(4) One node interchange between a customer on the artificial route and a regular route (insertion).

(5) Two node interchange between routes on the same day.

(6) Two node interchange between routes on different days.

(7) Two node interchange between a route and the artificial route.

In each of these seven cases, it is important to indicate whether each of the customers involved in the interchange belongs to M or M'.

We examine each customer for each of the seven node interchanges in which he might participate for possible positive trade-offs in the distribution cost. Each of the node interchanges has a different expression that computes the savings in the cost. We present these savings expressions in the order that the list of node interchanges was presented. We denote the savings by OS.

(1) If a customer changes routes on the same day then the inventory quantity delivered to him does not change. Following the illustration of a one-node interchange in Figure 2.2 the savings is given by

$$(3.1)\quad OS(i,j;\, k,r,\ell) = v_{ij} + v_{kr} + v_{r\ell} - v_{jr} - v_{ri} - v_{k\ell}$$

This OS term expresses the savings (in \$) if customer r switches routes. His original routing position is between customers k and ℓ and a subsequent routing position would be between customers i and j.

(2) If a customer who is originally assigned to a route on a given day switches positions to another route on a different day then the quantity delivered to him does change. Consequently, the cost of pumping this inventory into his tank differs and so are the cost coefficients. (The cost coefficients are the c_{ik}'s or the g_{ik}'s whatever is appropriate.) Assume that the customer r switches from day b to day f and belongs to the subset M. If we denote the pumping cost for customer i on day k by o_{ik} then the savings expression has the form :

$$(3.2)\quad OS(i,j;\, k,r,\ell;\, b,f) = v_{ij} + v_{kr} + v_{r\ell} - v_{k\ell} - v_{jr} - v_{ri} + c_{rb} - c_{rf} - o_{rb} - o_{rf}.$$

In the case when this customer r belongs to the subset M', the g_{rb} and g_{rf} replace the coefficients c_{rb} and c_{rf}.

(3) If a customer is deleted from the set of routes to the artificial route, he has to belong to the subset M'. Assume that the day he was scheduled for replenishment was day f. The savings expression is

$$(3.3)\quad OS(k,r,\ell;\, f) = v_{kr} + v_{r\ell} - v_{k\ell} + g_{rf} + o_{rf}.$$

(4) If a customer is inserted from the artificial route into a regular route, say on day b, between customers i and j, the savings expression is given by

$$(3.4)\quad OS(i,j;\, r;\, b) = v_{ij} - v_{jr} - v_{ri} - g_{rb} - o_{rb}.$$

(5) If two customers r and s assigned to two different routes scheduled on the same day interchange positions, then, following Figure 2.3 the savings expression has the following form

$$(3.5)\quad OS(i,r,j;\, k,s,\ell) = v_{jr} + v_{ri} + v_{ks} + v_{s\ell} - v_{kr} - v_{r\ell} - v_{js} - v_{si}.$$

(6) If two customers interchange positions on routes from different days b and f, then the savings expression has the following form

$$(3.6)\quad OS(i,r,j;\, k,s,\ell;b,f) = v_{jr} + v_{ri} + v_{ks} + v_{s\ell} - v_{kr} - v_{r\ell} - v_{js} - v_{si} + c_{rb} - c_{rf} + c_{sf} - c_{sb} + 0_{rb} - 0_{rf} + 0_{sf} - 0_{sb}.$$

(7) If two customers interchange positions where say, customer s switches from the artificial route into the position of customer r who was previously scheduled for replenishment on day b, then the savings expression has the form

$$(3.7)\quad OS(i,r,j;\, s;\, b) = v_{jr} + v_{r} - v_{js} - v_{si} + g_{rb} - g_{sb} + 0_{rb} - 0_{sb}$$

In the case that a customer will stock-out if inserted into a given route, the corresponding temporal cost in the OS expression for that customer is equal to the cost of a special delivery or other penalty cost, together with an appropriate sign. Not all favorable trade-offs--as indicated by a positive sign on the OS value--are feasible if we check the corresponding vehicles' capacities, and the drivers' time constraints for the proposed interchange.

We have to make sure that the vehicle capacity will not be exceeded and the driving constraint not violated when interchanging customers among the respective routes. We denote by d_{ik} customer's i demand on day k. By q^h we denote the capacity of the n^{th} vehicle. Thus, in the two-node

interchange the following conditions have to be satisfied before the customers i and j are swapped in their respective routes.

(i) The term OS has to be positive

(ii) $\sum_{s \in P_k^{hr}} d_{sk} - d_{ik} + d_{jk} \leq q^h$

and $\sum_{s \in P_k h'r} d_{sk'} - d_{jk'} + d_{ik'} \leq q^{h'}$

where k and k' correspond to the original days for customer i and j respectively.

For the one-node interchange, where customer i switches to route $p_{k'}^{h'r'}$, condition (ii) has only one expression : $\sum_{s \in P_{k'}^{h'r'}} d_{sk'} + d_{ik'} \leq q^{h'}$.

We deliberately ignore the tie constraints while testing for a feasible swap. We do this because we assume (and our computational results validated this assumption) that the vehicle capacity limitation will force any one route to terminate before the drivers' workday is over. If the vehicles do not differ in their capacity characteristics (i.e., if they're all the same size) this entails that we must solve a 'packing problem' that 'packs' routes efficiently onto the available vehicles for a specific route assignment. (In this way, we attempt to minimize the number of vehicles used on any particular day.)

4. The "Greedy Node Interchange Algorithm" and the "Matching Interchange Algorithm"

We now describe two interchange procedures. The first one we will call the "Greedy Interchange Algorithm" and the other one we will call the "Matching Interchange Algorithm." The initial input to both procedures is the set of vehicle routes for each day of the planning period which we have obtained as output from the second stage of the hierarchical optimization

approach. Each of the interchange procedures at some point calls up a single route improvement heuristic-(SRIH).

SRIH : Initialization : Input a single route.

Step 1 : Perform a 2-opt.arc interchange heuristic and go to Step 2.

Step 2 : Perform a 3-arc interchange improvement heuristic.

Test : Is the tour also 2-opt?

Yes : STOP.

No : Go to Step 1.

The Greedy Interchange Algorithm chooses the interchange with the highest savings term each time it performs an interchange transformation. When no more transformations that are feasible and have a positive savings term exist, the procedure attempts to reoptimize each route using SRIH procedure. The criteria for termination is reached when after reoptimizing each route no more feasible interchanges with positive savings exist.

The "Greedy Interchange Algorithm"

Denote by TR the total number of routes in a given planning period. Denote by TNUM(I) the name of the I^{th} route (I = 1,...,TR)

Step 1 : For I = 1,TR call SRIH(TNUM(I))

Step 2 : Compute all the OS (savings) terms for one-node and two-node interchange transformations. Together with each OS computation test if the OS value is positive, and if the interchange corresponding to the term is feasible. Drop the nonpositive and infeasible interchanges. Record only the positive and feasible OS terms. If there are no positive and feasible OS terms STOP.

Step 3 : Sort the OS terms in nonincreasing order.

Step 4 : If the list of the OS terms is not empty, perform the interchange transformation that corresponds to the first OS term. If the list is empty, go to Step 1.

Step 5 : Recompute the other OS terms on the list while dropping the nonpositive and infeasible terms. Go to Step 3.

The "Matching Interchange Algorithm"[1]

The obvious weakness of the "Greedy Interchange Algorithm" is that the Greedy interchanges in the beginning of the procedure may result in inferior quality (i.e., higher cost) routes at the end. The Matching Interchange Algorithm attempts to overcome that tendency of the greedy approach.

We will illustrate the competing approaches (Greedy and Matching) in the following example :

Example. Assume (for simplicity) only four interchanges with positive savings. The first three are two-node interchanges, and the fourth one is a one-node interchange

Interchanges		
Customer	Customer	Savings (in units)
1	2	9
2	3	3
1	3	10
-	3	9

We construct the following graph :

---

[1] A description and the mathematical formulation for the maximum weight matching problem are in [3] among many other references.

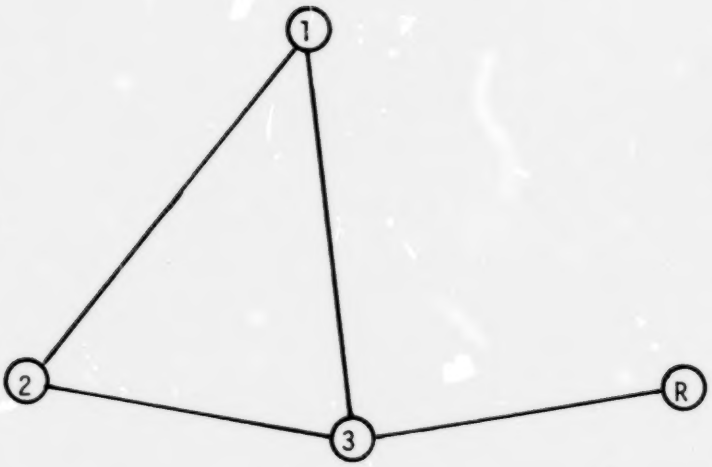

Figure 4.1 : Graph corresponding to the above example.

The node numbers correspond to the customer numbers. The node indicated by R corresponds to the route into which customer 3 can be inserted. The edges correspond to the interchanges between the customers or customer-switch to a different route. The numbers next to the edges correspond to the saving realized if this interchange is performed. If we solve this example by the greedy approach it would result in savings of 10 units; this is illustrated by the edge with double line in Figure 4.2.

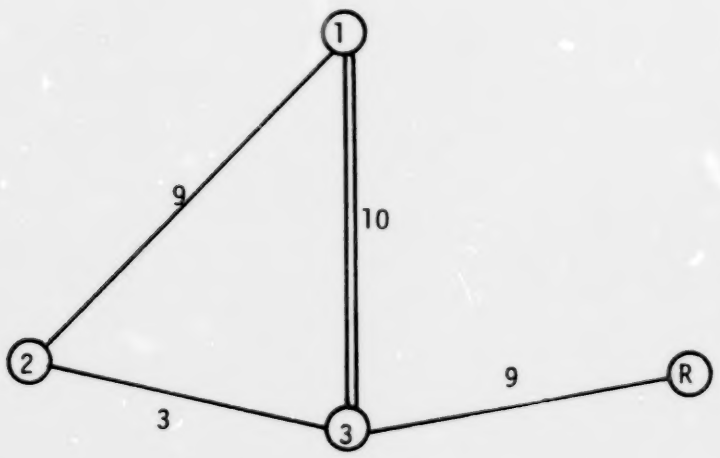

Figure 4.2 : Graph corresponding to the greedy solution.

Solving the same example by the matching approach produces savings of 18 units as illustrated in Figure 4.3.

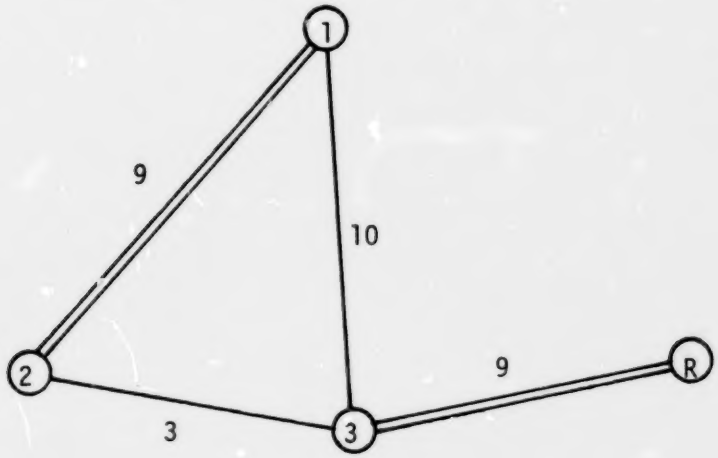

Figure 4.3 : Graph corresponding to the matching solution.

In the Greedy Interchange Algorithm we find the node interchange which results in the greatest improvement in the value of the objective function and implement this node interchange. We then, recompute from scratch all the node interchanges which are feasible and decrease the value of the objective. An interchange in this procedure is implemented one at a time. In the Matching Interchange Algorithm we attempt to implement several node interchanges simultaneously. In order for this procedure to present a valid route improvement, at each implementation of the node interchanges in the solution to the resulting matching problem, these interchanges have to be independent of each other, i.e., one node interchange cannot affect any other of the node interchanges in the formulation of the matching problem. This independence is insured in the Matching Improvement Algorithm and described in Step 6 of the procedure.

The Outline of the "Matching Interchange Algorithm"

Steps 1 and 2 of the second procedure are identical to the corresponding steps in the Greedy Interchange Algorithm. We start the description from Step 3.

Step 3 : For each route form a list of all the OS terms which correspond to an interchange resulting in a new customer coming into the route.

Step 4 : Find the largest element on each route list of OS terms.

Step 5 : If all route lists are empty

then Go to Step 8;

else Compare the maximal elements from each list and choose the one with the over all maximum value. (In the case of a tie, resolve it arbitrarily.) Denote the interchange corresponding to the maximum value by $\overline{\overline{OS}}$.

Step 6 : The interchange $\overline{\overline{OS}}$ effects at most four customers, not counting the ones switching positions. In the case of a two-node interchange, the four are the two pairs of customers preceding and succeeding the ones interchanged. In the case of a one-node interchange, one pair consists of the two customers between which another customer is inserted and the other pair consists of the two customers between which the customer is deleted. The case in which one of the affected routes is the artificial route has to be treated separately in this step. Delete from all the route lists the OS terms in which the effected customers are considered for an interchange.

Step 7 : Construct a graph G as follows. If the $\overline{\overline{OS}}$ corresponds to a two-node interchange, then each customer corresponds to a node in G and the edge joining these two nodes is assigned the value of

the $\overline{\overline{OS}}$ term. If the $\overline{\overline{OS}}$ term corresponds to a one-node interchange, then add one artificial node to G, and the other node corresponds to the customer changing routes. The edge connecting the two nodes is assigned the value of the $\overline{\overline{OS}}$ term. Delete the $\overline{\overline{OS}}$ term from the route lists and Go to Step 5.

Step 8 : Solve the maximum weight matching problem on graph G. For each route, form a list of interchanges in the solution to the matching problem that have a customer/node in common with the route.

Step 9 : Examine the feasibility of the routes obtained by implementing the interchanges from the solution to the matching problem. In some cases the capacity constraint of the resulting routes might be violated. If all routes are feasible

then Go to Step 1;

else Go to Step 10.

Step 10: Pick the infeasible route for which the demand most exceeds the capacity constraint. (If there is a tie, then break it arbitrarily.) Examine for this route the list of interchanges in the solution to the matching problem and delete from this list the ones that do not add demand to the route.

Notation. Denote the list of interchanges that add demand to the maximally infeasible route by : MIR. Denote by $\bar{d}_i$ the added demand for interchange $i \varepsilon$MIR and by $\bar{q}$ the excess demand on that route. OS_i denotes the saving value for the interchange i.

Step 12 : Solve[1] the 'sorting' problem

$$\text{minimize} \sum_{i \in MIR} OS_i y_i$$

(M1) $$\text{subject to} \sum_{i \in MIR} \bar{d}_i y_i > \bar{q}$$

$$y_i = 0 \text{ or } 1 \text{ for all } i$$

Delete all the interchanges i such that $y_i = 1$ in the solution to (M1) from the route (i.e., reverse that interchange transformation). Go to Step 9.

A simple illustration of a graph G that might be constructed in Step 7 is given in Figure 4.4 below. We assume only three routes R_1, R_2 and R_3 for this illustration.

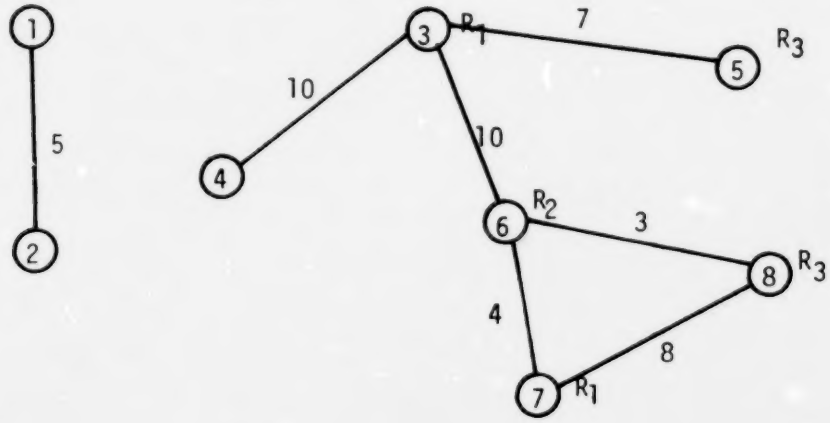

Figure 4.4 : An example of a graph constructed in Step 3 of the matching interchange algorithm.

---

1 Note : The 'sorting problem' has an equivalent Knapsack problem formulation. In Step 12 a fast heuristic solution scheme is implemented.

In Figure 4.4, nodes 2, 3, 5, 6, 7 and 8 represent customer nodes; nodes 1 and 4 are artificial nodes. The R_1, R_2, R_3 next to the nodes indicate the customers' route numbers. The number next to an edge indicates the savings value associated with that interchange transformation.

5. The Design of the Computational Experiments

In the actual computational experiments conducted in order to compare the performance of the Greedy Improvement Algorithm with the Matching Improvement Algorithm, two slightly different Matching improvement procedures were implemented.

One Matching improvement procedure, refered to here as the "Clean Matching", was implemented as described in the outline of the Matching Improvement Algorithm. The other Matching Improvement procedure, refered here as the "Greedy Matching", is implemented in three stages. In the first stage, the Matching Improvement Algorithm is applied for OS terms computed in Step 2 that exceed a certain value (in our case the first value chosen is 4.0).

The first stage is completed when there are no interchange savings (OS terms) of value greater than or equal to 4.0. In the second stage, the Matching Improvement Algorithm is applied for OS terms which exceed the value of 1.0. In the third and last stage, the Matching Improvement Algorithm is applied for all positive OS terms. The Greedy Matching procedure halts when an interchange with a positive OS term can no longer be found. The intention, when implementing the Greedy Matching procedure, was to save computer memory (up to 30 K for the problem tested), but it also turned out to be the scheme with the best routing solution in the majority of cases.

In this computational experiment, we used real data from an inventory routing problem as described in [1], [2], [7], [8] and [9]. A feasible routing solution for a planning period (i.e., the input solution to the improvement schemes) was constructed using a so-called Modified Vehicle Routing Algorithm which is described in detail in [7], and is based on a version of a Clark and Wright algorithm. In order to insure the same starting feasible routing solution on each of the 12 consecutive weekly periods for the three improvement schemes, only the Greedy Improvement solution was used to update the state of the customers in the system.

In Table 1 we present two types of weekly results for each route improvement procedure tested in this study. One result is simply the corresponding value of the objective function. The other result is given in units per hour and describes the efficiency of the routing solution in terms that are accepted in the heating oil distribution industry.

Wk	Feas. OBJ	Sol. Units/hr	Greedy OBJ	Match Units/hr	Clean OBJ	Match Units/hr	Greedy OBJ	Greedy Units/hr
1	549	3704	282	4423	281	4395	277	4494
2	433	4384	188	6011	190	6014	180	6092
3	496	4026	245	5648	250	5590	249	5689
4	527	3623	206	4836	202	4872	203	4852
5	447	4623	253	5666	245	5744	261	5616
6	704	4077	417	5405	434	5198	407	5370
7	470	4482	202	5537	206	5504	227	5223
8	582	4400	317	5374	318	5367	318	5366
9	619	3927	296	5106	299	5020	303	5049
10	398	4390	181	4958	177	4947	174	4967
11	376	4663	133	5411	140	5244	136	5383
12	528	3729	201	5508	225	5209	214	5298

Table 1

In Table 2 we present the percentage improvement in the distribution efficiency measure (i.e., in units/hour) resulting when implementing the three route improvement procedures.

	Units/hour		
Wk	Greedy Match % improvement	Clean Match % improvement	Greedy % improvement
1	19.4	18.7	21.3
2	37.1	37.2	39.0
3	40.3	38.8	41.3
4	33.5	34.5	33.9
5	22.6	24.2	21.5
6	32.6	27.5	31.7
7	23.5	22.8	16.5
8	22.1	22.0	22.0
9	30.0	27.8	28.6
10	12.9	12.7	13.1
11	16.0	12.5	15.4
12	47.7	39.7	42.1
Average % Improvement	28.14 (28.1)	26.53 (26.5)	27.20 (27.2)

Table 2

In Table 3 we present the percentage improvement in the weekly routing objective function when comparing the three route improvement procedures.

	OBJ		
Wk	Greedy Match % improvement	Clean Match % improvement	Greedy % improvement
1	48.6	48.8	49.5
2	56.6	56.1	58.4
3	50.6	49.6	49.8
4	60.9	61.7	61.5
5	43.4	45.2	41.6
6	40.8	38.4	42.2
7	57.0	56.2	51.7
8	45.5	45.4	45.4
9	52.2	51.7	51.1
10	54.5	55.5	56.3
11	64.6	62.8	63.8
12	61.9	57.4	59.5
Average % Improvement	53.05 (53.1)	52.40 (52.4)	52.57 (52.6)

Table 3

When we compare percentage improvement in the routing performance as reflected in Table 2 and Table 3 (i.e., in units/hour versus in the value of the objective) a valid question as to which of the two system performance measures one should relate is raised. The answer is that in an actual operational environment, as encountered for example in heating oil distribution systems, the performance measure used is units/hour. It actually represents how many commodity units the system delivers on average in one hour of operation. The maximization of this measure is usually our 'true' objective. The objective function used in our system is of surrogate nature. The comparison among the three route improvement schemes has to be based on the values of the objective function used.

Note : The three route improvement procedures tested in this experiment were coded without much emphasis on their performance with regard to running time. Our main objective has been to test concepts without investing heavily in algorithmic efficiency as measured by CPU time on the Univac 1108 which was used. Nevertheless, the running times for the procedures are quite good, and certainly their cost is affordable by most moderatly-sized distribution companies. For example, the "Greedy Improvement Algorithm" for a problem with 190 nodes required 19.7 CPU seconds, and 78 node problem, 8.4 seconds. The "Greedy Matching Improvement" required 25.0 CPU seconds for a 190 node problem, and 8.2 seconds for 78 a node problem.

6. Summary

Given a set of vehicle routes over a time period, we demonstrate the usefulness of heuristic route improvement scheme which are capable of examining and operating on all the routes simultaneously. We develop a node interchange operation on many routes similar to the arc exchange

transformation on a single route. This concept of node interchange operation on routes is instrumental in designing improvement shcemes with the flexibility of interchanging the positions of two nodes-customers, one node, or even deleting or inserting nodes into a vehicle routing solution. As was demonstrated for the inventory routing problem, the resulting improvement in the value of the objective function for all three schemes developed was between 52 and 53 percent. When interchanging the positions of customers on a route and between routes, we attempted to exploit each customer's temporal cost characteristics and special cost characteristics.

In the IRP illustration, given a planning period, the number of routes considered simultaneously ranged between 10 and 20, and the number of customers between 70 and 200.

Given an approximately uniform distribution of customers' locations over a bounded geographic area with the depot in its center, and a random partition-assignment of these customers into weekdays, a node interchange operation between initially constructed routes proved to be a very effective tool for redesigning an improved routing system.

The Greedy Improvement procedure can be considered as an extension of a procedure described in [14]. The difference is that our procedure considers not only single node interchanges, but also two node interchanges. It has the flexibility of deleting and inserting customers into the routing solution.

The Matching Improvement procedure attempts to adopt a more global node interchange approach. It is a procedure that in its improvement process constructs and solves a minimum weight matching problems. While conceptually attractive, it did not perform considerably better than the

Greedy Improvement procedure. [We note that] To our knowledge, it is the first time that a matching problem subroutine is used as a tool in a route improvement procedure. We demonstrated that with some care, a matching code can be incorporated successfully into a more elaborate vehicle routing heuristic. There is no indication of a clear superiority of one-node interchange improvement scheme over the others as applied to the IRP. All the three schemes presented perform very well and considerably improve the initial routing solution.

REFERENCES

[1] A. Assad, B. Golden, R. Dahl and M. Dror, "Design of an Inventory/-Routing System for a Large Propane-Distribution Firm", Proceedings of Southeastern TIMS Meeting, 315-320, October 82

[2] ———, ———, ———, ———, "Evaluating the Effectiveness of an Integrated System for Fuel Delivery", Proceedings of Southeastern TIMS Meeting (1983)

[3] M. Ball, U. Dergis, "An Analysis of Alternate Strategies for Implementing Matching Algorithms", to appear in Networks

[4] ———, L. Bodin, "A Computational Study Comparing the Efficiency of Various List Structures in Matching Algorithms", in preparation

[5] L. Bodin, B. Golden, A. Assad, M. Ball, "The State of the Art in Routing and Scheduling of Vehicles and Crews", Computers & OR (special issue) June 1983

[6] N. Christofides and S. Eilon, "Algorithms for Large-Scale Traveling, Salesman Problems", Operational Research Quarterly, 23, 511 (1972)

[7] M. Dror, "The Inventory Routing Problem", Ph. D. Thesis, College of Business and Management, University of Maryland at College Park (1983)

[8] ———, M. Ball, B. Golden, L. Bodin, "Inventory Routing Problem", submitted for publication

[9] ———, ———, "Inventory Routing : Reduction from an Annual to Short Period Problem" Centre de recherche sur les transports - Publication #354, University of Montreal, 1984

[10] A. Federgrnen and P. Zipkin, "A Combined Vehicle Routing and Inventory Allocation Problem", Research Working Paper No. 345 A, Graduate School of Business, Columbia University (June 1980)

[11] M. Fisher, R. Taikumar and W. Bell, "The Impact of Advanced Technologies on Inventory Management", Proceedings of the Annual Meeting NCPDM (Oct. 1981)

[12] S. Lin, "Computer Solutions to the TSP" Bell System Tech. J., 44, 22-45 (1965)

[13] ———, and B. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem", Operations Research, 21, 498-516 (1973)

[14] U. Tawsak, "The Swap Algorithm to Further Refine the Vehicle Dispatch and School Bus Problem", Ph. D. Thesis, University of Missouri-Rolla (1979)